

LISTING OF THE CLAIMS:

1. (Currently Amended) A method of processing a text file in a computer application, comprising the steps:

forming a template from fragments of the text file that include literals for an output text stream;

A2
using the template as an overlay for parsing incoming files, or as a prototype to generate a segment of an output file;

using a macro class to map data from the text file to an application; and

embedding the macro class as a keyword within the template, wherein when the template reaches the keyword, the template calls the macro class to further process the text file according to a given set of rules.

2. (Original) A method according to Claim 1, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

3. (Original) A method according to Claim 1, wherein the macro class derives data from the application and formats it into the text file.

4. (Original) A method according to Claim 1, wherein the macro class derives a template name from the invoking template and uses that name to invoke a next template to further process the text file.

5. (Original) A method according to Claim 1, further comprising the step of providing an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.

6. (Currently Amended) A system for processing a text file in a computer application, comprising:

means for forming a template from fragments of the text file that include literals for an output text stream;

means for using the template as an overlay for parsing incoming files, or as a prototype to generate a segment of an output file;

means for using a macro class to map data from the text file to an application; and

means for embedding the macro class as a keyword within the template, wherein when the template reaches the keyword, the template calls the macro class to further process the text file according to a given set of rules.

7. (Original) A system according to Claim 6, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

8. (Original) A system according to Claim 6, wherein the macro class derives data from the application and formats it into the text file.

9. (Original) A system according to Claim 6, further comprising an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.

10. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for processing a text file in a computer application, said method steps comprising:

forming a template from fragments of the text file that include literals for an output text stream;

using the template as an overlay for parsing incoming files, or as a prototype to generate a segment of an output file;

using a macro class to map data from the text file to an application; and

embedding the macro class as a keyword within the template, wherein when the template reaches the keyword, the template calls the macro class to further process the text file according to a given set of rules.

11. (Original) A program storage device according to Claim 10, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

12. (Original) A program storage device according to Claim 10, wherein the macro class derives data from the application and formats it into the text file.

13. (Original) A program storage device according to Claim 10, wherein said method steps further comprise the step of providing an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.